



Busca en Internet información sobre estas cuestiones:

- A tu disposición tienes un algoritmo de KMeans como un ejemplo de un algoritmo de agrupamiento, en el que tienes descrito sus campos.
 - En 'n_clusters' como número de agrupaciones de los datos.
 - En 'max_iter' como número de ejecuciones del algoritmo por cada centroide.
 - En 'tol' como medida de la tolerancia en la convergencia entre los centroides.
 - En 'verbose' como ejecución comentada.
 - En 'algorithm' como algoritmo que se empleará para hacer las agrupaciones.
- Cambia algunos de los valores en los campos anteriores y observa el resultado. Guarda varias imágenes de los cambios efectuados y del gráfico asociado.



Ilustración 1 Visualizacion Original



```
centros = 4
print( "Número de centros: ", centros )
X, y = make_blobs(
    n_samples = random.randint( 100, 125 ),
    #n_features = random.randint( 2, 10 ),
    centers = centros
)

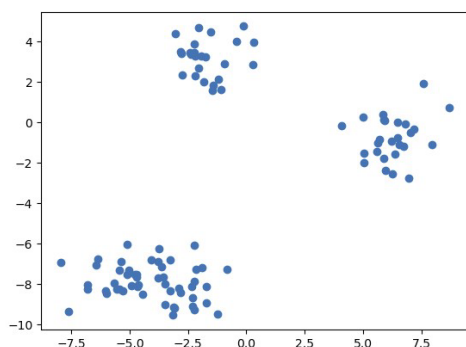
plt.scatter( X[ :, 0 ], X[ :, 1 ] )
plt.show()

#-----
# Algoritmo KMeans - Algoritmo de agrupamiento
#-----
# n_clusters número de agrupaciones (y centroides) a generar. (int, default = 8)
# init método de inicialización (k-means, random)
# n_init veces que se ejecuta el algoritmo con cada centroide (auto, int)
# max_iter máximo de veces que se ejecuta el algoritmo por ejecución (int, default = 300 )
# tol tolerancia en la convergencia entre los centroides (float, default = 1e-4)
# verbose ejecución comentada (int, default = 0)
# random_state generación de números aleatorios (con init es determinística). (int, instancia de RandomState, None)
# copy_x si los centroides se recalculan antes de empezar el funcionamiento. (bool, default = True)
# algorithm algoritmo k-means que se utilizará (lloyd, elkan, auto, full). Por defecto es lloyd.
#-----
agrupaciones = 5
print( "Número de agrupaciones: ", agrupaciones )
kmeans = KMeans(
    n_clusters = agrupaciones,
    max_iter = 5000,
    tol = 1e-7,
    verbose = 0,
    algorithm = "elkan"
)
kmeans.fit( X ) # X
y_kmeans = kmeans.predict( X )
plt.scatter( X[ :, 0 ], X[ :, 1 ], c = y_kmeans, cmap = 'viridis' )
plt.show()
```

✓ 0.3s

Ilustración 2 Cambios en algunos parametros

Número de centros: 4



Número de agrupaciones: 5

<c:\Users\dres\.conda\envs\Curso\lib\site-packages\sklearn\cluster.py>
super()._check_params_vs_input(X, default_n_init=10)

